Predictive RD Optimized Motion Estimation for Very Low Bit Rate Video Coding *

Faouzi Kossentini [1], Yuen-Wen Lee[1], Mark J. T. Smith [2], and Rabab Ward[1]

[1]Department of Electrical Engineering,
University of British Columbia,
Vancouver BC V6T 1Z4, Canada

[2]School of Electrical and Computer Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332-0250, USA

Abstract

Predictive rate-distortion (RD) optimized motion estimation techniques are studied and developed for very low bit rate video coding. Four types of predictors are studied: mean, weighted mean, median, and statistical mean. The weighted mean is obtained using conventional linear prediction techniques. The statistical mean is obtained using a finite-state machine modeling method based on dynamic vector quantization. By employing prediction, the motion vector search can then be constrained to a small area. The effective search area is reduced further by varying its size based on the local statistics of the motion field, through using a Lagrangian as the search matching measure and imposing probabilistic models during the search process. The proposed motion estimation techniques are analyzed within a simple DCT-based video coding framework, where an RD criterion is used for alternating between three coding modes for each 8 × 8 block: motion only, motion compensated prediction and DCT, and intra DCT. Experimental results indicate that our techniques yield very good computation-performance tradeoffs. When such techniques are applied to an RD optimized H.263 framework at very low bit rates, the resulting H.263 compliant video coder is shown to outperform the H.263 TMN5 coder in terms of compression performance and computations, simultaneously.

*This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant # OGP-0187668 and NASA.

1 Introduction

A variety of motion estimation algorithms have been developed for very low bit rate video coding. However, the block matching algorithm (BMA) [1] stands as the most popular and the simplest in concept, design, and implementation. In fact, many new BMA-based video compression algorithms allow transmission or storage of QCIF resolution video with acceptable quality at bit rates as low as 16 kilobits per second (kbps) [2, 3, 4, 5]. Most notable are the H.263-based video coders [6], which have recently been shown to outperform video coders using more complex object-based and model-based motion estimation algorithms. A two-step BMA-based motion estimation algorithm is adopted in many H.263-based video coder implementations such as Telenor's TMN5 [7]. The first step is an integer-pel accuracy full-search BMA (FS-BMA). The second step is aimed at improving the estimation accuracy, producing motion vector estimates with $\frac{1}{2}$-pel accuracy.

There are many problems associated with the above two-step motion estimation algorithm. The FS-BMA is well known for its large computation requirements, which has fueled many research activities. Such activities have produced more efficient algorithms [1, 3] such as log search, three-step search, cross search, conjugate gradient search, hierarchical search, and block subsampling. However, most of these algorithms may quickly get trapped in local minima, yielding a significant loss in motion vector estimation performance. Moreover, the FS-BMA performs poorly during intensity and reflectance illumination changes, non-translational motion activities such as zoom and rotation, scene changes, and occlusions. This, coupled with the FS-BMA's sensitivity to video input noise, produces a non-smooth motion field that costs many precious bits in very low bit rate video coding applications. Finally, producing motion vector estimates with $\frac{1}{2}$-pel accuracy increases both the complexity and the bit rate while yielding a relatively insignificant improvement in video quality.

In this paper, we present predictive rate-distortion (RD) optimized motion estimation techniques employing several predictors and search methods. The proposed techniques reduce substantially the number of computations, produce a smoother motion field and yield better reproduction quality, simultaneously. The techniques are analyzed and compared in the context of a simple DCT-based video coding framework, where only 8×8 blocks are used and only three coding modes (motion only, motion compensated prediction and DCT, intra DCT) are allowed. An RD criterion, expressed by the Lagrangian $D + \lambda R$, is used to alternate between the above coding modes.

In very low bit rate video coding applications such as video telephony, the motion field is very structured and slowly varying. Moreover, the motion vectors are usually limited in magnitude. This suggests that significant computation and coding gains can be achieved by taking advantage of the strong spatio-temporal dependencies that exist between the motion vectors. Memory has always been incorporated into the motion vector coding process, but very few researchers have suggested exploiting it to simplify motion estimation. Recently, predictive motion estimation [8, 9, 10, 11, 12, 13] has become an important research area. This paper studies the complexity and performance of two linear predictors (mean and weighted mean) and two non-linear predictors (median and statistical mean) when applied to motion estimation in the context of very low bit rate video coding. Conventional linear prediction techniques are used to obtain the mean and weighted mean. The statistical mean is estimated using conditional probabilities obtained via a finite-state machine (FSM) that is uniquely determined by a dynamic state vector quantization (VQ) codebook.

By employing prediction, we can restrict the motion vector search to a small area whose center is the predicted motion vector. To further reduce the search area size, we introduce two probabilistic models that place soft constraints on its size, allowing the area to expand or contract as a function of the local statistics of the motion field. Let's assume that the search area is divided into layers, as will be described later. The first model is based on the hypothesis that if the cost, expressed here in terms of a Lagrangian, increases when going from one layer to the next, then it is unlikely that we will find a better motion vector by continuing the search outward. The second model is much more constrained because it is based on the hypothesis that only when the cost increases monotonically over a sequence of three layers, that searching more layers for a better motion vector is not required. Because there are many areas when non-motion changes occur, a hard constraint is placed on the maximum number of layers that can be searched.

Another characteristic of very low bit rate video coding applications is that the motion vector bit rate occupies a substantial portion (as much as 50%) of the total bit rate. RD optimized motion estimation algorithms have been introduced [14, 17, 18] [15, 16, 11, 19, 12] that reduce the motion vector bit rate substantially. Another attractive solution is to code in a non-lossless fashion the motion vectors using, for example, VQ [20, 15, 11] or statistical methods [16, 11]. In this paper, we build on our earlier work [15, 16, 11, 19, 12], where we employ a Lagrangian (matching error biased by motion vector bit rate) as the search matching measure. Given specific constraints, such a measure yields the best RD tradeoffs.

Our motion estimation techniques are simple and computationally efficient, yet their estimation performance is comparable to that of the FS-BMA, even when the Lagrangian $D + \lambda R$ is minimized in both cases. When the proposed techniques are applied to an H.263 RD optimized framework, the resulting H.263 compliant very low bit rate video coder outperforms Telenor's best TMN5 coder in terms of both computations and quality. Our video coder also has the important advantage that the quality, bit rate, and complexity are easily controllable. In what follows, we present the proposed motion estimation techniques. This is followed by a description of our DCT-based video coding framework. Section 4 briefly describes the RD optimized H.263 framework used for the development of the H.263 compliant video coder. The last two sections present our experimental results and some conclusions, respectively.

2 Proposed Motion Estimation Techniques

We here develop motion estimation techniques where the spatio-temporal statistical dependencies within the motion field are effectively exploited. Our objective is mainly to ease the computational burden, although our techniques also improve the RD tradeoffs and reduce the estimation's sensitivity to video input noise. Before we describe the techniques, several parameters must be defined. These parameters are the motion estimation block size, the initial search area size, motion vector accuracy, and the number of motion vectors per block. For simplicity, we set the block size to 8 × 8. Such a block size leads to good RD tradeoffs within the DCT-based framework described in Section 3. We also set the size of the initial square search area to ±16 pixels. Since the application of our techniques is very low bit rate video coding, sub-integer pixel accuracy estimates are neither required nor helpful. Finally, as B-frames are used in our framework, only one motion vector per block is estimated.

Now, suppose that the video frame targeted for motion estimation is partitioned into 8 × 8 blocks. For each block, a vector $\mathbf{d} = (x, y) \in \mathcal{S}$, where $\mathcal{S}$ is the set of all possible vectors in a variable search area, is sought that minimizes the Lagrangian

$$J_{\lambda}^{M}(\mathbf{d}) = \sum_{\mathbf{r} \in \mathcal{W}} (I(\mathbf{r}, n) - I(\mathbf{r} + \mathbf{d}, n-1))^2 + \lambda\, R^{M}(\mathbf{d}), \tag{1}$$

where $\mathbf{r}$ is the spatial index of the image pixels, n is the time index, $I(\mathbf{r}, n)$ is the image intensity at position $\mathbf{r}$ of the candidate block in the current frame, $I(\mathbf{r} + \mathbf{d}, n-1)$ is the image intensity at position $\mathbf{r} + \mathbf{d}$ of the matching block in the previous frame, $\mathcal{W}$ is the size of the matching window, and $R^{M}(\mathbf{d})$ is the motion vector bit rate [1]. Minimizing the Lagrangian $R^{M}(\mathbf{d})$ is

[1] Note that only self-information estimates are computed during the Lagrangian minimization.

guaranteed only by considering all possible candidate motion vectors in the search area. The computations involved in one search operation are squaring and adding together the differences in the matching window $\mathcal{W}$ between the pixels of the current block and those of the candidate block, and then adding the result to a biased estimated rate. The cost of each search operation can be made smaller by either reducing the size of the matching window or by performing sub-sampling techniques [1]. Either of these methods can lead to a significant reduction in estimation performance. The method used in this work is the partial Lagrangian computation approach, which is a generalization of the method described in [21]. A motion vector candidate is rejected once the accumulated Lagrangian becomes larger than the current minimum value. The term $\lambda\ R^M(\mathbf{d})$ is first computed and compared to the minimum Lagrangian obtained so far. If $\lambda\ R^M(\mathbf{d})$ is larger, the motion vector candidate is rejected. Otherwise, the same test is applied sequentially as the squared pixel differences are added, until either the accumulated Lagrangian is larger than the minimum or all pixel differences within the window have been added. This method can reduce the number of computations required for the minimization of $J_\lambda^M(\mathbf{d})$ by as much as 80% at very low bit rates. Unfortunately, however, the computational load is still high, especially when the search area is large (e.g., ±64). The obvious solution is to minimize the size of the search area without sacrificing a significant loss in estimation performance.

By accurately predicting the location of the best motion vector candidate, one can then search a relatively small area in the neighborhood of the predicted motion vector, while still locating the "optimal" motion vector. This is indeed possible, especially in very low bit rate video applications, owing to the large amount of motion field redundancies within the same frame as well as between consecutive frames. Figure 1 illustrates two examples of a motion vector search area. Since there is usually a larger motion activity in the horizontal direction, the search area shown in Figure 1(b) may be more suitable. However, experimental results show that the difference in the resulting performance is often insignificant. Our proposed method is to first estimate the most likely integer motion vector $\mathbf{v} = (x_i, y_i)$ given a prediction model. Then, only the candidate motion vectors in one of the small diamond-shaped search areas whose center is $\mathbf{v}$ is considered. The x and y components of $\mathbf{v}$ are the closest integers to the corresponding (generally *real*) components of the predicted motion vector.

2.1 Prediction

Assuming a fixed search area, we next discuss four different predictors: mean, weighted mean, median, and statistical mean. The design of the prediction parameters is simplified by computing correlation coefficients and mutual information values between motion vectors within a sufficiently large 3-D region of support (ROS) and the current motion vector. Such a region includes previously coded motion vectors representing blocks that are close spatially and/or temporally. Figure 2 shows a 3-D ROS extending over two frames. Block G of the previous frame is positioned at the same spatial location as the unlabeled block in the current frame (or current block). The labels are the average correlation coefficients (c_x, c_y) (top) and average mutual information values (m_x, m_y) (bottom) of the x and y components between the ROS motion vectors and the current one (i.e., the motion vector of the current block). Note that these values decrease rapidly as we go away from the current block along the spatio-temporal axis, and that spatial dependencies are stronger than temporal ones.

Mean Predictor

The mean predicted motion vector is given by $\tilde{\mathbf{v}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{v}_k$, which is the average of the K motion vectors $\mathbf{v}_k$, evaluated for each x and y component independently. As confirmed by our experimental results, only the vectors corresponding to the closest blocks (A,B,D) in Figure 2 should be used. Including poorly correlated motion vectors in the computation of the average adversely affects prediction accuracy.

Weighted Mean Predictor

A more accurate predicted vector is given by $\tilde{\mathbf{v}} = \sum_{k=1}^{K}\alpha_k\mathbf{v}_k$, where the linear prediction coefficients α_k's are closely related to the correlation coefficients shown in Figure 2. In this work, the α_k's are computed off-line using the well-known autocorrelation method. Based on Figure 2, only the motion vectors representing the blocks (A,B,C,D) are significantly correlated with the current motion vector. Thus, a fourth order predictor is used. A higher order predictor improves slightly the prediction accuracy, but at the expense of a large increase in number of computations. While this weighted mean predictor outperforms the uniform mean predictor discussed above, further improvements in performance are expected by changing the values of the coefficients α_k's adaptively during encoding.

Median Predictor

Like those of the mean and weighted mean predicted vectors, the two components of the median predicted vector are computed independently but using the same procedure. Two different median predicted vectors are computed: one based on the H.263 3-block ROS (A,B,D) and another based on a 5-block ROS (A,B,C,D,G). The two ROS's are shown in Figure 3. When used in our simple DCT-based framework, the two ROS's performed similarly. As will be shown later, the 3-block median predictor is better in terms of prediction performance. However, the 5-block median predictor has the additional advantage that it can more effectively hide channel errors.

Statistical Mean Predictor

A more powerful non-linear predictor is the statistical mean vector $\tilde{\mathbf{v}} = (\tilde{x}, \tilde{y})$, whose two components are also computed independently and in an identical manner. Without loss of generality, we will next describe the procedure used to estimate $\tilde{x}$, the x component of $\tilde{\mathbf{v}}$. The scalar $\tilde{x}$ is the weighted average of x components of all possible motion vectors. The weighting coefficients are equal to the conditional probabilities of the x components. The conditional probabilities are determined using a finite-state machine (FSM) model that is represented by a codebook, or a set of state code vectors. Each code vector represents a relatively large set of template vectors $\mathbf{u_t}$. The components of the template vectors $\mathbf{u_t}$ are feature symbols representing the x components of previously coded motion vectors in the ROS. In other words, a template vector $\mathbf{u_t}$ is the output of a mapping function whose inputs are the ROS motion vector x components, and is the input to a vector quantizer (VQ) whose codebook contains all state code vectors. In this work, the following two mapping functions and VQs are considered:

1. Scalar quantized mean: the mean of the x components is computed and scalar quantized. The ROS associated with the x components is the one used in the computation of the mean predicted vector. Each possible value of the scalar quantized mean represents a template vector $\mathbf{u_t}$.

2. Scalar quantized weighted mean: the input to the scalar quantizer is the weighted mean, computed using the same ROS adopted during the computation of the weighted mean predicted vector.

The state VQ codebook is generated on-line [11]. The codebook is initialized with one code vector $\mathbf{u}_s^1$, the first template vector. The next template vector $\mathbf{u}_t$ is compared to $\mathbf{u}_s^1$. If the distortion $d(\mathbf{u}_t, \mathbf{u}_s^1)$ given by

$$d(\mathbf{u}_t, \mathbf{u}_s^1) = \sum_{i=1}^{N} |u_t^i - u_s^i|,$$

is less than a threshold T_1, then $\mathbf{u}_t$ is mapped to the state code vector $\mathbf{u}_s^1$. Otherwise, $\mathbf{u}_t$ is added to the codebook. Similarly, each new vector $\mathbf{u}_t$ is compared to all vectors in the codebook. If the best matching code vector is still not a good match, as determined by the threshold T_1, then it is added to the codebook. When the maximum codebook size is reached, the least popular code vector is deleted before adding the new vector.

Next, suppose $\mathbf{u}_t$ is the template vector whose components are the different feature symbols considered above. Then, an N-size state codebook $\mathbf{C} = \{\mathbf{u}_s^1, \mathbf{u}_s^2, \ldots, \mathbf{u}_s^N\}$ is searched, and the probability table corresponding to the state code vector $\mathbf{u}_s^*$ closest to $\mathbf{u}_t$ is selected. Then, $\tilde{x}$ is given by

$$\tilde{x} = \sum_{\ell=1}^{L_x} p(x_\ell | \mathbf{u}_s^*) x_\ell, \qquad (2)$$

where $p(x_\ell | \mathbf{u}_s^*)$ is the probability of the motion vector x component x_ℓ given state $\mathbf{u}_s^*$ and L_x is the number of all possible motion vectors.

Since both the codebook and the probability tables are adaptive, this procedure is expected to yield very accurate statistical mean predicted motion vectors. Note that our implementation of the statistical mean predictor does not require that the table probabilities be stored or even directly processed. The estimates can be obtained easily by maintaining a counter for each state code vector. Thus, such a prediction technique is relatively simple.

To summarize, the prediction performance clearly depends on the model parameters such as the ROS, the estimation accuracy of the prediction coefficients, the bit rate of operation and the content of the video scene. While FSM-based prediction is most accurate (as shown in Section 5), it is worth noting that even for less accurate predictors such as the uniform mean or the median, it is found that the FS-BMA yields at most 5% of the motion vectors that do not belong to the diamond-shaped search area shown in Figure 1(a). This should not be surprising, since many of the low resolution video sequences such as MISS AMERICA exhibit very small/slow motion and non-motion related variations.

2.2 Search Area Size

The size of the diamond-shaped search area can be expressed in terms of contours or layers, as shown in Figure 1. Each layer represents many possible motion vectors. The number of vectors per layer increases as a function of the distance from the center (layer 0 in Figure 1(a)) of the search area. The number of layers that must be searched depends greatly on the required accuracy of the predictor. A conceptually simple technique is to search a pre-determined fixed number of layers, where the number is chosen such that a good balance between average number of required computations and estimation performance is achieved. However, this technique can be inefficient, since if the prediction is accurate, the motion vector located at the center is likely the best candidate. One solution is to halt the search if the Lagrangian J_0, associated with selecting and encoding the center motion vector, is relatively small. That is, if $J_0 < T_2$, where T_2 is a fraction of the current average Lagrangians associated with some previously coded neighboring motion vectors, then the search can be safely stopped. Otherwise, all the pre-specified layers are searched. The computation-performance tradeoffs depend greatly on the threshold T_2. If T_2 is too large, estimation performance can deteriorate. If it is too small, many layers will have to be searched, which requires a large number of computations. Thus, T_2 should be selected adaptively given, for example, a constraint on the allowed number of computations.

The above technique works well when motion vectors can be accurately predicted. A good example is the well-known MISS AMERICA video sequence, where the background stays nearly constant as a function of time. Moreover, with the exception of the eye and lip movements, the non-zero motion vectors are very structured. In general, however, motion vectors can only be approximately predicted, as motion can be fast and complex. Nevertheless, where motion-related changes occur, the motion vectors can still be expected to be localized in the neighborhood of the predicted vector.

Building on this notion of motion localization, we next introduce an alternative technique, where we employ a probabilistic model that places a soft constraint on the size of the diamond-shaped search area. First, let us assume that the search area is divided into layers, following the configuration shown in Figure 1(a). We also assume that the layers $0, 1, 2, \ldots$ are searched sequentially in the same order (i.e., layer 0, then layer 1, etc.) as we go away from the center of the search area. Finally, let $J_0, J_1, J_2, \ldots$ be the minimum Lagrangians associated with the layers $0, 1, 2, \ldots$, respectively. We next propose two probabilistic models, where each model is based on a specific hypothesis. The two hypotheses are:

- Hypothesis I: If J_n of layer n is larger than J_{n-1} of layer $n-1$, then it is unlikely that we will find a better motion vector by continuing the search outward. Thus, searching more layers is not necessary. This hypothesis is often violated in areas where the motion field is not smooth. In such a case, the Lagrangian surface is likely not convex. Even when the motion field is smooth, the search as specified by this hypothesis can quickly get trapped in a local minimum.

- Hypothesis II: Only if $J_{n-1} < J_n < J_{n+1}$, will we be confident that searching more layers for a better motion vector is wasteful. This hypothesis places a larger convexity constraint on the search. It almost guarantees optimality, but at the expense of a much larger computational load. Because this test may never be satisfied, we must limit the number of considered layers to, for example, 16.

By using any of the two models corresponding to the above two hypotheses, we allow the search area to expand or contract based on the local statistics of the motion field. Either model can be better than the other in terms of computation-performance tradeoffs, depending on the particular video sequence

being coded. But the first model (Hypothesis I) would be the choice when computational complexity is a major concern. As can be expected, both models fail in areas where many non-motion changes occur. Finally, note any of the two models can be used interchangeably, and the models can also be used in conjunction with the above threshold-based technique, as dictated by performance and/or complexity constraints.

3 DCT-Based Video Coding Framework

The proposed motion predcition and search techniques are studied, analyzed, and tested within a simple RD optimized DCT-based framework using only 8×8 vectors and involving only intra or forward inter coding of the luminance Y component. For simplicity, the two chrominance pictures are not coded. The first video frame is coded using the JPEG baseline coder. The following frames in the same group are predictively coded. Predictive coding consists of motion only coding, motion compensated prediction and DCT residual coding, or intra DCT coding.

As described in the previous section, each motion vector is obtained by searching a diamond-shaped area whose center is the predicted motion vector and size is a variable that depends on the local statistics of the motion field. The motion vector is then represented by a vector offset with respect to the predicted motion vector. The entropies of the x and y components of the vector offset are used to estimate the motion vector bit rate. After determining the motion vector $\mathbf{d}^*$ leading to the minimum value of the Lagrangian $J_\lambda^M(\mathbf{d})$, the corresponding 8×8 difference block is coded using essentially the same residual coder implemented in [7]. The only difference is that the QUANT parameter $q_R \in Q_R$ is chosen to minimize the Lagrangian

$$J_{\lambda,\mathbf{d}^*}^R(q_R) = D_{\mathbf{d}^*}(q_R) + \lambda \, R_{\mathbf{d}^*}(q_R), \tag{3}$$

where $R_{\mathbf{d}^*}(q_R)$ and $D_{\mathbf{d}^*}(q_R)$ are the DCT coder's average bit rate and mean squared error (MSE), associated with $q_R \in Q_R$. Next, the original 8×8 block is intra coded by minimizing the Lagrangian

$$J_\lambda^I(q_I) = D(q_I) + \lambda R(q_I), \tag{4}$$

where $R(q_I)$ and $D(q_I)$ are similarly the bit rate and MSE, associated with parameter $q_I \in Q_I$ of the intra DCT coder. Finally, let the quantity R_m be the bit rate associated with specifying the mode m, where $m = 1$ represents motion only, $m = 2$ represents motion compensated prediction and DCT, and $m = 3$ represents intra DCT. By incorporating such information, and other types of side information (e.g., quantizer number), the three Lagrangian values are computed, and the mode leading to the smallest value is selected for the current 8×8 block.

Clearly, achieving the best RD performance can only be guaranteed by computing the three Lagrangian values. This is computationally expensive, mainly because all 32 QUANT parameter values would have to be considered. Fortunately, however, the average bit rates and MSEs do not have to be computed for all QUANT values. In fact, even after exhaustive searching, only a few QUANT values are selected most of the time, as is indicated by the skewed distribution shown in Figure 4. Thus, very little loss in performance is sacificed when only the $2-4$ most likely QUANT values are considered.

4 RD Optimized H.263 Framework

The above DCT-based framework simplifies the study and performance evaluation of the various motion estimation techniques developed in this work. However, to illustrate the potential computation-performance advantages of such techniques, we develop an H.263 compliant RD optimized video coder where 1) motion vectors representing lumninance 16×16 blocks (Y-MBs) are obtained, macroblocks[2] (MBs) are motion compensation predicted, and the corresponding

[2] Each macroblock consists of one Y-MB and two chrominance 8×8 blocks.

8×8 difference blocks are DCT coded. The coder employs the simple H.263 2-D median predictor, a motion search algorithm based on Hypothesis I, and an RD criterion for the selection between H.263's macroblock (MB) coding modes. Details about the RD optimized H.263 framework can be found in [12]. We next briefly describe the MB coding mode selection criterion. This is followed by a discussion of our RD and computation control methods, which are more efficient than the ones developed in [12].

4.1 MB Coding Mode Selection Criterion

During intra coding of I-pictures, only the intra DCT coding mode is allowed. Our intra DCT coder is similar to the one implemented in [7]. The only difference is that our selection of a value for the quantizer paramater QUANT (5 bits) is based on an RD criterion. More specifically, QUANT of the first MB is set to 16, and QUANT of each other MB is set during the encoding process to the previous MB's QUANT value, which is likely adjusted by one of the 4 possible values of DQUANT (2 bits). The value of DQUANT (if any) is selected such that the Lagrangian $J_I(\lambda) = D_I + \lambda R_I$, where R_I and D_I are the bit rate and MSE, is minimized.

During inter coding of P-pictures, the ideal MB coding mode selection method would be to compute six Lagrangians, each corresponding to one of the six H.263 principal modes, and choosing the mode leading to the smallest Lagrangian value. The six principal modes are:

SKIP: The MB is skipped, and the COD parameter is set to 1.

INTER: One motion vector and the corresponding DCT coefficients are coded.

INTER+Q: Same as INTER except that QUANT is possibly changed.

INTER4V: Similar to INTER except that four motion vectors are coded.

INTRA: The 8×8 blocks of the original MB are DCT coded.

INTRA+Q: Same as INTRA except that QUANT is possibly changed.

Although motion vector estimation and DCT coding are performed independently, and the QUANT value is predicted (only 4 values are considered), computing all *six* Lagrangian values for each MB is generally impractical. Our approach to reducing the required number of computations is to employ thresholding techniques that allow us to safely eliminate the expensive INTRA and/or INTER coding options from consideration. Details about the H.263 RD optimized MB coding mode selection method can be found in our other paper [12].

4.2 RD and Computation Control

For different Lagrangian parameter values λ's, different RD tradeoffs can be obtained. To analyze and evaluate our search and prediction techniques within the DCT-based framework, a heuristical procedure is used to estimate appropriate values for λ. For a fair comparison with the TMN5, however, the proposed H.263 coder must employ an algorithm that adaptively provides an accurate estimate of the value of λ given rate and/or distortion constraints. The value of λ can be estimated following the methods described in [22, 23, 17, 24]. Such methods, however, either are iterative or require that accurate models be developed for the input video signal. In this work, we propose an alternative method where the parameter λ is updated during the encoding process using a new recursion formula. Without loss of generality, let's assume that we have a fixed-rate communication system that is governed by $s(t+1) = s(t) + R(t) - B$, where t denotes time, $s(t)$ is the size of the buffer, $R(t)$ is the variable output bit rate of the encoder, and B is the fixed output rate of the buffered contents. The parameter λ is initially estimated based on computed long-term statistics, and is then varied adaptively based on the linear model described by the equation $\lambda(t) = c\ s(t)$. Assuming that $s^* = \frac{S_{max}}{2}$ is the desired buffer size, where S_{max} is the maximum physical buffer size, we can write $\lambda^* = cs^*$. Thus, another expression for $\lambda(t)$ is given by $\lambda(t) = \lambda^* \frac{s(t)}{s^*}$. Then, if we recursively apply the formula given by

$$\lambda(t) = \lambda(t-1)\, \frac{s(t)}{s^*},$$

we should hopefully converge to the fixed point λ^*. In practice, updating λ based on the above recursion formula is found to be an efficient solution. In fact, relative to the more complex algorithm described in [12], this algorithm is shown experimentally to perform quite well. In particular, using a buffer of size 10 kilobits, the problem of overflow/underflow was never encountered during our coding simulations.

The Lagrangian parameter λ does impact the computation demands of the video coder. For example, a large value of λ can reduce the number of computations substantially. However, such number can be precisely controlled by appropriately selecting values for T_1, T_2, and other threshold parameters. Given an explicit constraint, such as maximum number of adds/multiplies, the threshold parameters can be increased or decreased so that the average number of computations is close enough to the imposed constraint. An efficient recursive algorithm similar to the one described above can be used for this purpose.

5 Experimental Results

5.1 Introduction

The target bit rates for our experiments are in the range between 4 and 16 kbps for color sequences. The MISS AMERICA and CAR PHONE sequences in QCIF format at 10 frames per second are selected for testing. As stated earlier, only integer-pel accuracy motion estimation is used. For fairness, the MSE cost measure is also used in TMN5's implementation of motion estimation. Next, the prediction techniques discussed in this paper are evaluated in terms of performance and complexity. This is followed by an analysis of the two search models defined by Hypotheses I and II. Both the evaluation and analysis are performed using our simple DCT-based framework. A comparison between our H.263 RD optimized video coder and the TMN5 is provided at the end of this section.

5.2 Prediction

Tables 1 and 2 show the average entropy for MISS AMERICA and CAR PHONE (respectively) of the motion vector x component in the search area centered at the mean, weighted mean, median, and statistical mean. There are two different mean predictors: (1) MEAN-A where the ROS in Figure 2 consists of blocks (A,B,D) and (2) MEAN-B where the ROS consists of blocks (A,B,C,D). As expected, MEAN-A leads to a lower entropy, as averaging many motion vectors adversely affects prediction accuracy. Since MEAN-A and MEAN-B are special cases of the weighted mean predictor (WM), the latter should yield better prediction accuracy. This is confirmed by the results shown in the tables. It is clear that the 3-block (A,B,D) median predictor (MED-A) outperforms the 5-block (A,B,C,D,G) median predictor (MED-B) in terms of both prediction accuracy and complexity. With the exception of its potentially higher robustness to channel errors, MED-B does not seem to be a good choice. When used to drive the FSM model, the weighted mean (WM-SM) also outperforms the uniform mean (M-A-SM and M-B-SM) in terms of prediction accuracy. Moreover, the statistical mean predictor (WM-SM) leads to a slightly better performance than WM. Finally, notice that MED-A yields a lower entropy than the more complex linear and statistical mean predictors. This, coupled with its higher robustness to channel errors, is likely what made it part of the H.263 standard.

It is clear from Tables 1 and 2 that the differences in average entropies are relatively small. This suggests that motion estimation/coding is not sensitive to prediction accuracy. Moreover, as shown in Tables 3 and 4 for MISS AMERICA and CAR PHONE (respectively), prediction-based motion estimation/coding compares favorably with statistical FSM-based estimation/coding studied in [15, 16, 11]. In the latter case, an FSM model based on dynamic VQ is employed for probability-based estimation and direct coding of the motion vectors (i.e., explicit prediction is not performed).

5.3 Search Area Size

Given a fixed prediction model, Figures 5 and 6 suggest that the two search models (represented by Hypotheses I and II) discussed above lead to more than one order of magnitude reduction in number of computations. The price paid is a relatively small loss in PSNR performance. For the sequence MISS AMERICA, the simpler model (Hypothesis I) yields better computation-performance tradeoffs. However, the two models achieve similar tradeoffs for the more active sequence CAR PHONE. It is nonetheless obvious that the two probabilistic models provide a significant advantage over the FS-BMA. Moreover, the model based on Hypothesis I appears to be the better alternative. Finally, besides the computational advantage, an important feature of this approach is that computational resources are better allocated, depending on the statistics of the input video sequence.

5.4 Video Coder

Figure 7 shows a comparison in terms of average PSNR between the our H.263 coder and the TMN5 (using all advanced options) of 150 frames of the MISS AMERICA sequence in the bit rate range of interest. Our coder differs from the TMN5 in that it employs median-based predictive RD optimized motion estimation based on the proposed techniques and RD optimized MB coding mode selection. Our coder performs significantly better than the TMN5, especially at the lower bit rates. Not only the PSNR is higher, but the subjective quality is also superior. For example, we presented many viewers with 150 frames of the decoded color sequence MISS AMERICA for several bit rates between 4 and 10 kbps. All the viewers reported that the subjective quality of our coder is noticeably higher.

Not illustrated in the figure is the fact that using the H.263 3-block median for prediction and applying Hypothesis I during the search process yields little or no loss in PSNR performance as compared to the case when the FS-BMA is used. A clear advantage of the new prediction and search techniques, however, is that the number of computations is reduced by approximately 20 : 1. Using our "C" implementation of the resulting coder, encoding requires 10 – 40 % of the time required by the TMN5.

6 Conclusions

We have studied and developed motion estimation techniques for very low bit rate video coding. The techniques are analyzed within a simple DCT-based video coding framework, where a simple RD criterion is used for alternating between three coding modes for each 8×8 block: motion only, motion compensated prediction and DCT, and intra DCT. First, the complexity and performance of four different predictors (mean, weighted mean, median, and statistical mean) have been evaluated. The most important results are that (1) motion estimation and coding is not very sensitive to prediction accuracy and (2) the H.263 3-block median predictor is a good choice when taking into consideration complexity, performance, and robustness to channel errors. Second, two probabilistic models (e.g., Hypotheses I and II) are imposed that allow the contraction or expansion of the search area depending on the statistics of the motion field. Our experimental results have shown that this approach generally results in more than one order of magnitude reduction in number of computations, while sacrificing an insignificant loss in PSNR performance.

The proposed motion estimation techniques are also applied to an RD optimized H.263 framework. Simulation results indicate that at very low bit rates, the resulting video coder significantly outperforms the TMN5 video coder both in terms of reproduction quality and number of computations, simultaneously. An additional advantage of our coder is that the bit rate and quality can be controlled through employing a simple iterative updating algorithm for the Lagrangian parameter λ. The computation requirements can also be controlled by appropriately selecting values for the threshold parameters.

References

[1] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards: Algorithms and Architecture.* Boston: Kluwer Academic Publishers, 1995.

[2] K.-H. Tzou, H. G. Musmann, and K. Aizawa, "Special Issue on Very Low Bit Rate Video Coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 4, pp. 213–367, June 1994.

[3] B. Girod, D. J. LeGall, M. I. Sezan, M. Vetterli, and Y. Hiroshi, "Special issue on image sequence compression," *IEEE Trans. on Image Processing*, vol. 3, Sept. 1994.

[4] W. Li, Y.-Q. Zhang, and M. L. Liou, "Special Issue on Advances in Image and Video Compression," *Proc. of the IEEE*, vol. 83, pp. 135–340, Feb. 1995.

[5] D. Anastassiou, "Current status of the MPEG-4 standardizaton effort," in *SPIE Proc. Visual Communications and Image Processing*, vol. 2308, pp. 16–24, 1994.

[6] ITU Telecom. Standardization Sector Study Group 15, "Document LBC-95 Working Party 15/1 Question 2/15," *Draft Recommendation H.263*, Leidschendam, 7 April 1995.

[7] Telenor Research, "TMN (H.263) encoder/decoder, version 1.4a," *TMN (H.263) codec*, May 1995.

[8] S. Zafar, Y. Zhang, and J. Baras, "Predictive block-matching motion estimation for TV coding— Part I: Inter-block prediction," *IEEE Transactions on Broadcasting*, vol. 37, pp. 97–101, Sept. 1991.

[9] Y. Zhang and S. Zafar, "Predictive block-matching motion estimation for TV coding— Part II: Inter-frame prediction," *IEEE Transactions on Broadcasting*, vol. 37, pp. 102–105, Sept. 1991.

[10] R. Arminato, R. Schafer, F. Kitson, and V. Bhaskaran, "Linear predictive coding of motion vectors," in *Proceedings of the IS&T/SPIE EI'96*, Jan. 1996.

[11] F. Kossentini, W. Chung, and M. Smith, "Rate-distortion-constrained subband video coding," *Submitted to Transactions on Image Processing*, Mar. 1996.

[12] Y. Lee, F. Kossentini, R. Ward, and M. Smith, "Towards MPEG4: An improved H.263-based video coder," *Accepted for publication in the MPEG4 Special Issue on Image Communication*, Aug. 1996.

[13] Y. W. Lee, R. K. Ward, F. Kossentini, and M. J. T. Smith, "Very low rate DCT-based video coding using dynamic VQ," in *ICIP96*, (Lauzanne, Switzerland), Sept. 1996.

[14] B. Girod, "Rate-constrained motion estimation," in *SPIE Proc. Visual Communications and Image Processing*, vol. 2308, pp. 1026–1034, 1994.

[15] W. Chung, F. Kossentini, and M. Smith, "A new approach to scalable video coding," in *IEEE Data Compression Conference*, (Snowbird, UT, USA), pp. 381–390, Mar. 1995.

[16] W. Chung, F. Kossentini, and M. Smith, "Rate-distortion constrained statistical motion estimation for video coding," in *ICIP95*, vol. 3, (Washington, DC), pp. 184–187, Oct. 1995.

[17] T. Wiegand, M. Lightstone, D. Mukherjee, T. Campbell, and S. Mitra, "Rate-Distortion Optimized Mode Selection for Very Low Bit Rate Video Coding and the Emerging H.263 Standard," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 182–190, Apr. 1996.

[18] D. Hoang, P. Long, and J. Vitter, "Efficient cost measures for motion compensation at low bit rates," in *IEEE Data Compression Conference*, (Snowbird, UT, USA), pp. 102–111, Apr. 1996.

[19] W. Chung, F. Kossentini, and M. Smith, "An efficient motion estimation technique based on a rate-distortion criterion," in *ICASSP96*, vol. 4, (Atlanta, GA, USA), pp. 1926–1929, May 1996.

[20] Y. Y. Lee and J. W. Woods, "Motion vector quantization for video coding," *IEEE Trans. on Image Processing*, vol. 4, pp. 378–381, Mar. 1995.

[21] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Boston: Kluwer Academic Publishers, 1992.

[22] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Englewood Cliffs, New Jersey: Prentice-Hall, 1983.

[23] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. on Image Processing*, vol. 2, pp. 160–174, April 1993.

[24] J. Choi and D. Park, "A stable feedback control of the buffer state using the Lagrangian multiplier method," *Special Issue on Image Sequence Compression*, vol. 3, pp. 546–558, Sept. 1994.

List of Tables

List of Figures

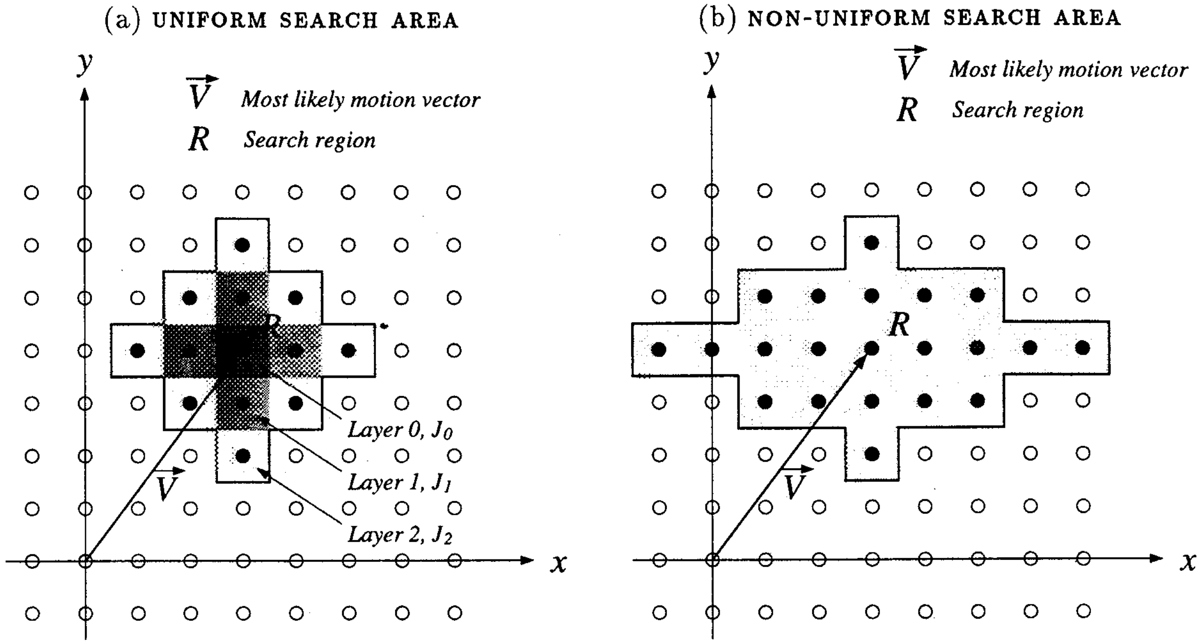


Figure 1: Two examples of a motion vector search area.

PSNR	MEAN-A	MEAN-B	WM	MED-A	MED-B	M-A-SM	M-B-SM	WM-SM
35.9	0.566	0.569	0.566	0.561	0.582	0.566	0.570	0.566
36.8	0.620	0.635	0.602	0.612	0.628	0.620	0.635	0.602
37.8	0.689	0.679	0.670	0.659	0.679	0.689	0.677	0.668

Table 1: Entropy for MISS AMERICA of motion vector x component offsets in the search area centered at the mean (MEAN-A and MEAN-B), weighted mean (WM), median (MED-A and MED-B), mean-based statistical mean (M-A-SM and M-B-SM), and weighted mean-based statistical mean (WM-SM).

PSNR	MEAN-A	MEAN-B	WM	MED-A	MED-B	M-A-SM	M-B-SM	WM-SM
31.2	1.686	1.737	1.598	1.553	1.602	1.658	1.681	1.588
33.0	2.008	2.044	1.930	1.860	1.928	1.946	2.011	1.909
34.9	2.318	2.335	2.190	2.109	2.194	2.237	2.296	2.169

Table 2: Entropy for CAR PHONE of motion vector x component offsets in the search area centered at the mean (MEAN-A and MEAN-B), weighted mean (WM), median (MED-A and MED-B), mean-based statistical mean (M-A-SM and M-B-SM), and weighted mean-based statistical mean (WM-SM).

		(0.139, 0.153) R		
	(0.140, 0.168) M	(0.161, 0.186) I	(0.141, 0.168) N	
(0.125, 0.136) Q	(0.151, 0.186) H	(0.166, 0.218) G	(0.160, 0.201) J	(0.153, 0.152) S
	(0.148, 0.146) L	(0.179, 0.166) K	(0.163, 0.159) O	
		(0.161, 0.145) P		

		(0.341, 0.376) F	
	(0.385, 0.395) C	(0.520, 0.512) B	(0.426, 0.377) D
(0.301, 0.279) E	(0.488, 0.487) A		

		(0.113, 0.083) R		
	(0.115, 0.087) M	(0.139, 0.107) I	(0.115, 0.085) N	
(0.094, 0.063) Q	(0.128, 0.098) H	(0.146, 0.128) G	(0.130, 0.098) J	(0.092, 0.060) S
	(0.114, 0.083) L	(0.137, 0.105) K	(0.111, 0.083) O	
		(0.104, 0.080) P		

		(0.208, 0.165) F	
	(0.240, 0.175) C	(0.328, 0.258) B	(0.231, 0.168) D
(0.164, 0.119) E	(0.286, 0.241) A		

Figure 2: Average correlation coefficients (c_x, c_y) (top) and average mutual information values (m_x, m_y) (bottom) between the ROS motion vectors and the current one.

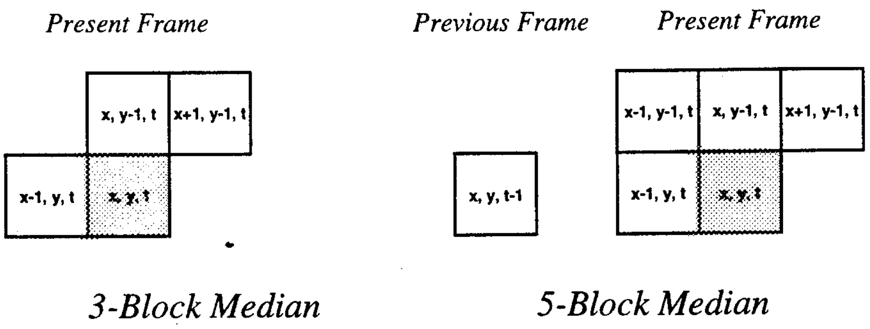


Figure 3: The two ROS's used to compute the median.

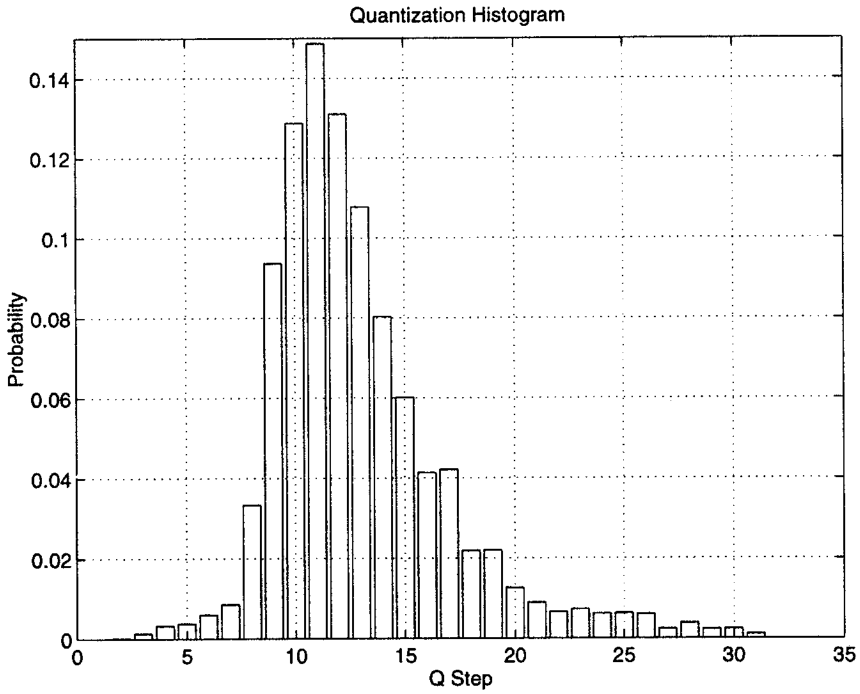


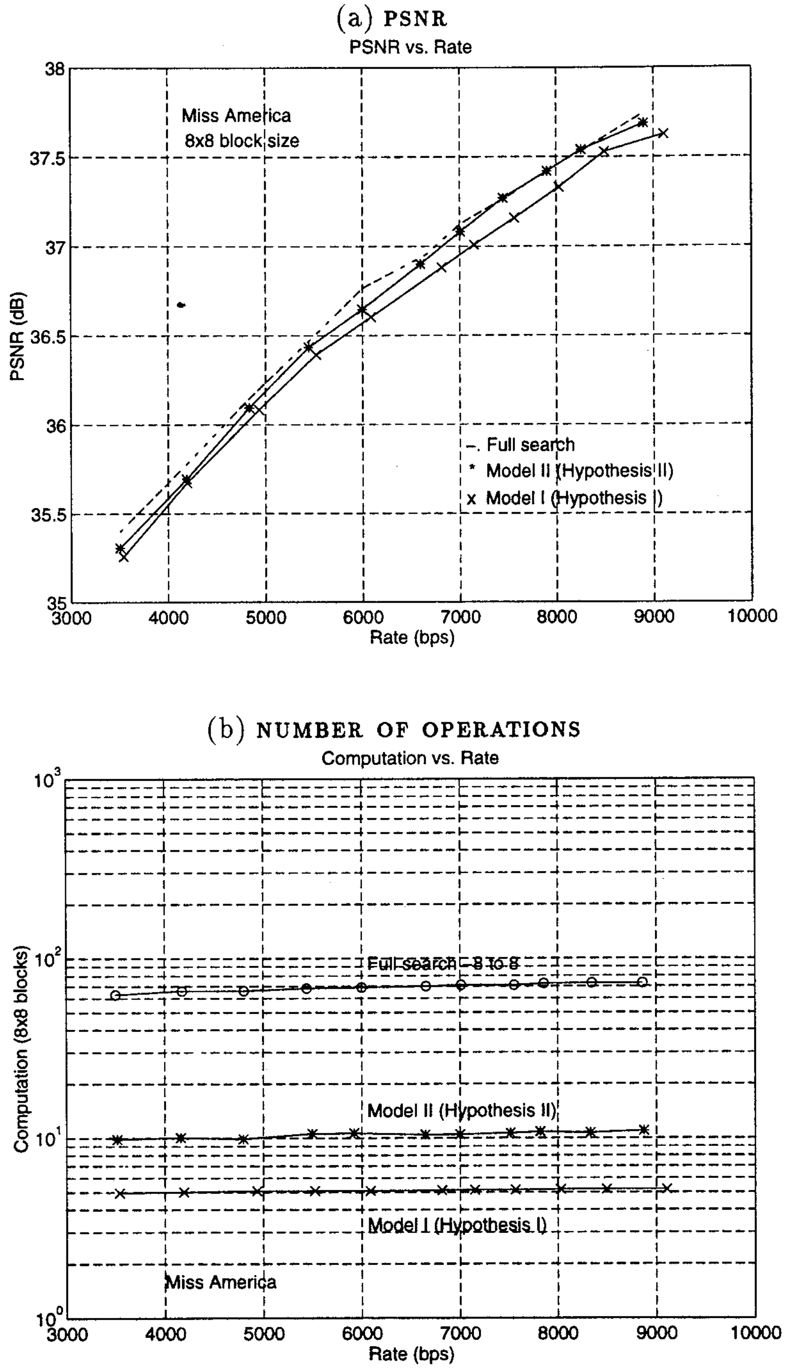
Figure 4: Probabilities of the quantizer matrices.

PSNR	WM-SM	FSM
35.9	0.569	0.498
36.8	0.602	0.547
37.8	0.668	0.592

Table 3: Comparison between the FSM (FSM) entropy of the motion vector x components and entropy of the x component offsets of those motion vectors located in the search area centered at the weighted mean-based statistical mean (WM-SM) for the sequence MISS AMERICA.

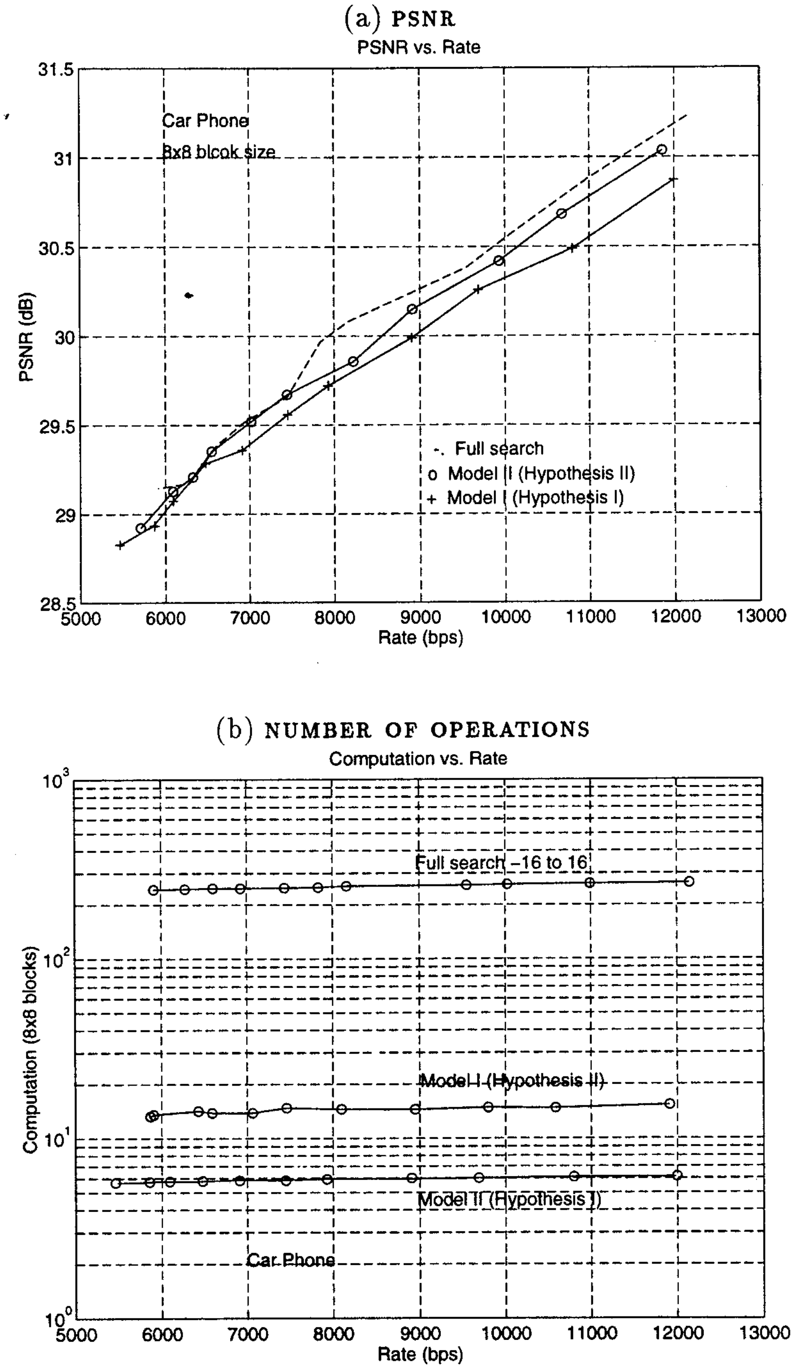
PSNR	WM-SM	FSM
31.2	1.598	1.469
33.0	1.930	1.751
34.9	2.190	2.026

Table 4: Comparison between the FSM (FSM) entropy of the motion vector x components and entropy of the x component offsets of those motion vectors located in the search area centered at the weighted mean-based statistical mean (WM-SM) for the sequence CAR PHONE.


(a) PSNR

(b) NUMBER OF OPERATIONS

Figure 5: Performance of the probabilistic models in terms of PSNR and number of computations relative to the FS-BMA: MISS AMERICA.



(a) PSNR

(b) NUMBER OF OPERATIONS

Figure 6: Performance of the probabilistic models in terms of PSNR and number of computations relative to the FS-BMA: CAR PHONE.

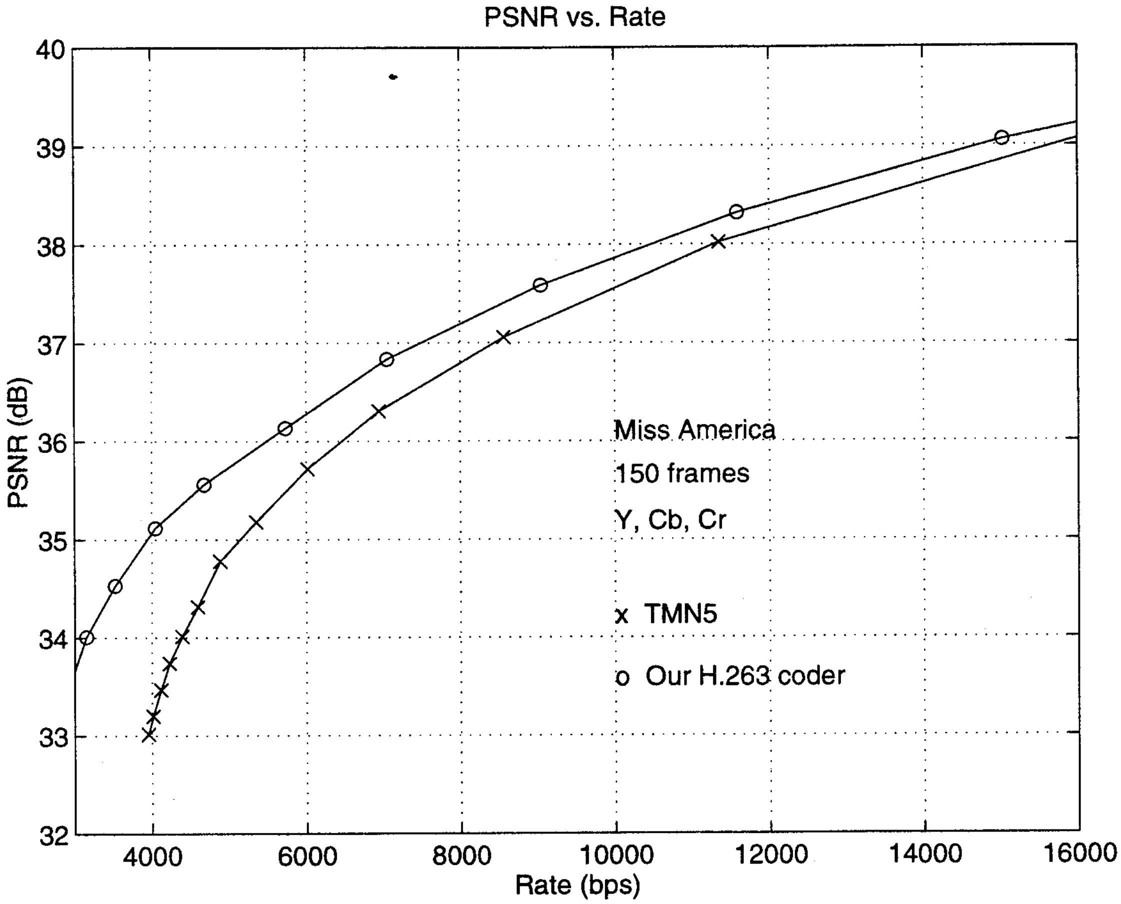


Figure 7: PSNR comparison between our coder and the TMN5 (all advanced options, MSE in motion estimation) for the test sequence MISS AMERICA at bit rates between 3 and 16 kbps. The comparison is based on the average PSNR of 150 color frames at 10 frames/sec.